

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
КАФЕДРА МЕХАНИКИ УПРАВЛЯЕМОГО ДВИЖЕНИЯ

ХАКИМОВ АНВАР ФАНИСОВИЧ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

РАЗРАБОТКА СИСТЕМЫ УПРАВЛЕНИЯ УДЕРЖАНИЕМ БПЛА НА
РАСЧЕТНОЙ ТРАЕКТОРИИ

Направление 010300

Фундаментальная информатика и информационные технологии

Научный руководитель,
кандидат физ.-мат. наук,
доцент
Шиманчук Д.В.

Санкт-Петербург

2016

Содержание

1	Введение	2
2	Постановка задачи	4
3	Обзор литературы	5
4	Задачи поиска оптимального управления	6
4.1	Подходы к решению задач поиска оптимальных управлений.	6
4.2	Описание алгоритма формирования управляющей силы . . .	7
4.3	Определение управляющих функций	9
5	Математическая модель	11
5.1	Описание математической модели	12
6	Описание алгоритма	16
7	Реализация алгоритма	19
7.1	Описание использованных средств	19
7.2	Результаты моделирования	20
8	Выводы	27
9	Заключение	28
10	Список литературы	29
A	Приложение	30

1 Введение

Под беспилотным летательным аппаратом (в дальнейшем БПЛА) обычно понимают любой летательный аппарат без экипажа на борту. Основными его задачами обычно являются воздушная съёмка и наблюдение в режиме реального времени за различными объектами. Однако в настоящее время БПЛА приобретают всё большую популярность для решения широкого круга задач гражданских и военных сфер. Основные предпосылки для этого следующие:

- Необходимость в относительно недорогих самоуправляемых летательных аппаратах для решения широкого круга задач гражданского и военного сектора;
- Появление достаточно дешёвых БПЛА, повлекло их распространение как части массовой продукции;
- Высокая сложность разворачивания необходимой инфраструктуры по месту требования для использования летательных аппаратов с пилотами.

Вообще говоря, БПЛА являются лишь частью сложного многофункционального комплекса. Можно выделить следующие элементы входящие в данную систему:

- Сам БПЛА;
- Наземная система управления;
- Необходимое программное обеспечение;
- Линии передачи данных;
- Дополнительные устройства, необходимые для выполнения задач.

Несложно понять, что управление БПЛА оператором, что в наше время в основном производится вручную, может повлечь большое количество аварийных ситуаций, чем если бы управление производилось в автоматическом, либо полуавтоматическом режиме. Такие режимы управления позволили бы избежать непредвиденных ситуаций, таких, как нестабильность,

либо полная недоступность линий связи и некорректных действий оператора. Поэтому изучение вопроса разработки системы позволяющей БПЛА производить движение по некоторой заранее заданной траектории является необходимым для качественной эволюции систем управления БПЛА.

Для автономного функционирования БПЛА содержит целый комплекс различных элементов. В основном это:

- Управляющая аппаратура;
- Трех-осевой гироскоп и акселерометр;
- ГЛОНАСС/GPS - приёмник;
- Датчики давления и воздушной скорости;
- Радио-модем.

В качестве управляющей аппаратуры, как правило, используются специализированные вычислители на базе цифровых сигнальных процессоров или компьютеры формата PC/104, MicroPC под управлением операционных систем реального времени (QNX, VME, VxWorks, XOberon). Программное обеспечение пишется обычно на языках высокого уровня, таких как C, C++ или Ada95.

Различные датчики используются для поддержки и корректировки полёта по заданному курсу. Кроме того они дают возможность быстро определить положение летательного аппарата в пространстве.

Кроме того, можно выделить любительские сборки БПЛА, в которых система управления представляется процессорными платами Arduino и Raspberry Pi под управлением различных дистрибутивов линукса, таких как Raspbian и Pidora (Fedora для Raspberry Pi).

2 Постановка задачи

Основной задачей данной научно-исследовательской работы является разработка компьютерной модели, которая моделируя полёт беспилотника, смогла бы задать БПЛА управления, необходимые для прохождения им определённого маршрута. Под определённым маршрутом в данной работе будет иметься ввиду рассчитанная по исходным данным траектория.

Для решения поставленной задачи необходимо выполнить следующие этапы:

1. Предварительно исследовать подходы к решению задач управления;
2. Описать математическую модель полёта летательного аппарата;
3. Определить алгоритм нахождения расчетной траектории;
4. Построить метод нахождения управлений БПЛА необходимых для достижения движения по необходимой траектории;
5. Синтезировать алгоритм для моделирования полёта БПЛА в соответствии с полученной моделью;
6. Реализовать Simulink-модель, моделирующую передвижение БПЛА по расчетной траектории;
7. Провести исследование адекватности полученной модели.

3 Обзор литературы

Достаточно подробное описание модели летательного аппарата, метода построения пространственной траектории движения и способа расчёта соответствующих управлений можно найти в [1]. Описание приёмов использования концепции обратных задач динамики с примерами и обширную библиографию по данному вопросу можно найти в [2], [3]. В [4] приведено описание построения t -параметрических траекторий в соответствии с используемой моделью. В частности представлен расчет t -параметрической траектории, связанной с планированием маневра смены эшелона. Кроме того показано, что данные траектории являются допустимыми.

В [5] можно найти описание БПЛА как объекта управления и некоторые требования к его управлению. Кроме того, в данной работе разработана имитационная динамическая модель БПЛА по упрощенной модели на основе действующих на летательный аппарат моментов.

В [6] можно найти документацию по использованию пакета прикладных программ MATLAB, среды имитационного моделирования Simulink и 3D Animation Toolbox.

4 Задачи поиска оптимального управления

4.1 Подходы к решению задач поиска оптимальных управлений.

Для начала сформулируем классическую постановку задачи поиска оптимальных управлений. Пусть управляемая система представлена математической моделью движения в виде дифференциального уравнения.

$$\dot{x}(t) = f(x, u), \quad (1)$$

где x - вектор координат состояния, u - вектор управляющих функций. Будем полагать, что векторная функция $f(x, u)$ - непрерывна и непрерывно дифференцируема по совокупности переменных. Сформируем задачу поиска оптимальных управлений. В момент времени $t = t_0$ система находится в положении, которое определяется вектором $x(t_0) = x_0$. Необходимо найти управление u такое, что они переводят систему в положение $x(t_1) = x_1$, а по траектории движения необходимо минимизировать следующий функционал:

$$J(u) = \int_{t_0}^{t_1} G(x, u) dt$$

В настоящее время разработаны различные пути решения задачи оптимального управления, такие как процедуры на основе принципа максимума, метода функций Ляпунова, динамического программирования и классического вариационного исчисления. Однако формальная постановка задач управления не всегда является оправданной и пригодной для практического применения. Причинами этого обычно является то, что:

- Во первых, действительные уравнения движения нередко известны только приближенно;
- Во вторых, высока сложность построения оптимизируемого функционала, которые, к тому же, нередко сложны для практического использования;
- И более того, даже если первые две проблемы разрешимы, возникают

ситуации, когда решить поставленную задачу невозможно.

По данным причинам в современной науке происходит поиск других способов конструирования алгоритмов управления движениям по принципам оптимальности. В настоящее время для решения задач автоматического управления широко используется концепция обратных задач динамики. В ней управляющие алгоритмы предполагается конструировать из условия того, что будут обеспечиваться предписанные динамические характеристики создаваемых систем. В классической постановке задачи, центральным является тот факт, что необходимо минимизировать функционал поиском управляющих функций $u(t)$, которые физически являются мерами перемещения управляющих механизмов объекта. В предлагаемом же подходе, данную задачу заменяют на задачу поиска таких управляющих сил, которые способны привести систему в требуемое движение. Нахождение управляющих сил в данном случае как раз производится в результате решения обратной задачи динамики при условии, что движение будет производиться по заранее известной траектории. Синтез алгоритма управления выполняется следующим образом:

1. Создание алгоритма, который формирует управляющую силу;
2. Определение управляющих функций, которые могут создать необходимые управляющие силы.

4.2 Описание алгоритма формирования управляющей силы

Рассмотрим способ построения алгоритма управления на примере системы с одной степенью свободы. Пусть математическая модель управляемого процесса представлена следующим уравнением:

$$\ddot{x}(t) + r(x)\dot{x}(t) + h(x)x(t) = f(x, \dot{x}, u),$$

где $f(x, \dot{x}, u)$ - нормированная по массе сила, x - вектор, определяющий положение системы, u - управляющая функция, $r(x)$, $h(x)$ - некоторые непрерывные функции.

Пусть в момент времени $t = 0$ справедливы следующие начальные условия: $x(0) = x_0, \dot{x}(0) = \dot{x}_0$. Необходимо определить такой алгоритм формирования управляющей силы, что система произведет движение из точки (x_0, \dot{x}_0) в начало координат по траектории:

$$x(t) = x^*(t), 0 \leq t < T < \infty \quad (2)$$

Предположим, что траектория $x^*(t)$ описана следующим равенством:

$$x^*(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}, \quad (3)$$

где λ_1, λ_2 - различные действительные или комплексно-сопряженные числа с отрицательной действительной частью, c_1, c_2 - постоянные определяемые начальными условиями. Отметим, что для системы (3) выполняется

$$x^*(t) \rightarrow 0, \dot{x}^*(t) \rightarrow 0, t \rightarrow 0$$

Что значит, что траектория $x^*(t)$ удовлетворяет краевым условиям. Поэтому, если существует управляющая сила, реализующая это движение, то она удовлетворяет условиям поставленной задачи. Данную силу можно найти по следующему отношению:

$$f^*(t) = \left(\frac{d^2}{dt^2} + r(x) \frac{d}{dt} + h(x) \right) x^*(t) \quad (4)$$

В [2] показано, что после подстановки (3) в (4) и некоторых преобразований связанных с учетом условий движения по траектории можно получить интересующий нас закон управления силой в следующем виде:

$$f^*(x, \dot{x}) = (r(x) + \lambda_1 + \lambda_2) \dot{x} + (h(x) - \lambda_1 \lambda_2) x \quad (5)$$

Важно отметить, что в каждый момент времени значения этого выражения формируются по текущим состояниям системы x, \dot{x} , а значит на основе обратной связи. Кроме того несложно изменить условия задачи для нахождения управляющей силы, которая переводит систему в окрестность

уже произвольной точки $(x^0, 0)$ за конечное время, если задать траекторию движения в следующем виде:

$$x^0 - x^*(t) = c_1 e^{\lambda_1 t} + c_2 e^{\lambda_2 t}$$

Однако форму задания траектории можно задавать и в другом виде. Например как решение дифференциального уравнения или значения полинома, который удовлетворяет краевым условиям.

4.3 Определение управляющих функций

Для регулирования управляющей силы $f(x, \dot{x}, u)$ необходимо определить управляющую функцию u , которая создает необходимое значение силы. Пусть параметры управляемого объекта известны и управляющая сила дана в виде (5). Согласно [2], данная сила однозначно определяется значением управляющей функции в каждый момент времени. Поэтому для каждого состояния системы $x(t), \dot{x}(t)$ можно указать значение силы u^* такое, что выполняется:

$$f(x, \dot{x}, u^*) = f^*(x, \dot{x})$$

Понятно, что для определения u^* нужно решить данное уравнение. Тогда если существует обратная функция, то решения можно представить в следующем виде:

$$u^*(t) = f^{-1}(f^*) \quad (6)$$

Заметим, что для уменьшения погрешности между $x(t), x^*(t)$ можно реализовать управление с обратными связями. Пусть модель системы возможно описать уравнением следующего вида:

$$\ddot{x} = f(x, \dot{x})u$$

и задана траектория (2). Будем искать управление u в форме обратных связей. Определим отклонения траектории следующим образом:

$$y = x - x^*(t) \quad (7)$$

Пусть данное отклонение соответствует управлению и подчиняются следующему условию:

$$\ddot{y} + k_1 \dot{y} + k_2 y = 0 \quad (8)$$

Заметим, что для такого отношения может выполняться следующая теорема. [7]

Теорема 1 *Для системы:*

$$\dot{X}(t) = A * X(t)$$

точка покоя системы асимптотически устойчива тогда и только тогда, когда действительная часть собственных чисел матрицы A меньше нуля.

Поэтому при выборе констант k_1, k_2 так, чтобы они удовлетворяли условию данной теоремы, можно гарантировать выполнение следующих условий:

$$x(t) \rightarrow x^*(t), \dot{x}(t) \rightarrow \dot{x}^*(t), t \rightarrow +\infty$$

Выполним подстановку (7) в (8) с учетом того, что

$$\ddot{x}^*(t) = f(x^*, \dot{x}^*)u^*, \ddot{x}(t) = f(x, \dot{x})u,$$

где u^* можно найти из соотношения (6). Получим следующее:

$$x(t)u - x^*(t)u^* + k_1(\dot{x}(t) - \dot{x}^*(t)) + k_2(x(t) - x^*(t)) = 0;$$

и далее получим управление в форме обратных связей:

$$u = \frac{x^*(t)u^* - k_1(\dot{x}(t) - \dot{x}^*(t)) - k_2(x(t) - x^*(t))}{x(t)}$$

5 Математическая модель

Перечень условных обозначений

u	Вектор управлений
δ	Общее время маневра
γ	Угол крена
ψ	Угол курса
ϑ	Угол наклона траектории
g	Ускорение свободного падения
H	Высота
k_1, k_2	Коэффициенты стабилизации
L	Продольная дальность
n_x	Продольная перегрузка
n_y	Поперечная перегрузка
t	Текущее время маневра
t_0	Начальное время маневра
u_p	Вектор программных управлений
V	Путевая скорость
y	Вектор состояния объекта
y_p	Вектор программных состояний объекта
Z	Боковая дальность

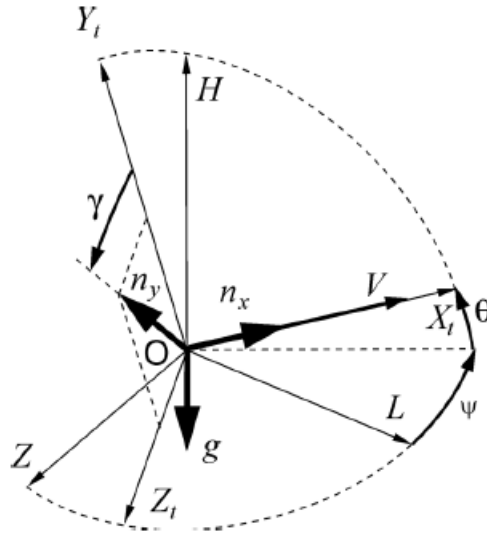


Рис. 1: Траекторная система координат.

5.1 Описание математической модели

Для начала определим системы координат в которых будет существовать модель. Для нормальной земной системы координат $OXYZ$ начало O расположим в произвольной точке земной поверхности. Ось OY направим противоположно направлению силы тяжести по местной вертикали. А оси OX, OZ расположим так, чтобы они составляли с осью OY правую систему координат. Предположим, что для решения данной задачи, ориентация системы координат является постоянной, и для небольших интервалов времени система инерциальна.

Далее получим нормальную систему координат O_tXYZ . Для этого перенесем начало нормальной земной системы координат в центр масс БПЛА. После этого получим траекторную систему координат $O_tX_tY_tZ_t$. Начало системы координат находится в центре масс БПЛА, ось O_tX_t направим по вектору скорости летательного аппарата. Ось O_tY_t направим там, чтобы она была расположена в местной вертикальной плоскости, которая содержит вектор скорости, направлена вверх и перпендикулярна оси O_tX_t . Ось O_tZ_t расположим так, чтобы она дополняла систему до правой системы координат. Будем считать, что расположение траекторной системы координат относительно нормальной задается углами наклона траектории θ и пути ψ . Данная система координат изображена на рис. 1.

Для описания движения БПЛА будем использовать следующую модель, описание которой можно найти в [1]. Модель представлена системой шести дифференциальных уравнений:

$$\begin{cases} \dot{V} = (n_x - \sin \vartheta)g \\ \dot{\vartheta} = \frac{(n_y \cos \gamma - \cos \vartheta)g}{V} \\ \dot{\psi} = -\frac{(n_y \cos \gamma - \cos \vartheta)g}{V \cos \vartheta} \\ \dot{H} = V \sin \vartheta \\ \dot{L} = V \cos \vartheta \cos \psi \\ \dot{Z} = -V \cos \vartheta \sin \psi \end{cases}$$

где V - путевая скорость, ϑ - угол наклона траектории, ψ - угол курса, H - высота, L - продольная дальность, Z - боковая дальность, n_x - продольная перегрузка, n_y - поперечная перегрузка, γ - угол крена, g - ускорение свободного падения. Продольная дальность, высота и боковая дальность представляют собой координаты положения центра масс БПЛА в нормальной земной системе координат. В качестве управлений рассматриваем перегрузки n_x, n_y и угол крена γ .

С помощью замены:

$$u_1 = n_x, u_2 = n_y \cos \gamma, u_3 = n_y \sin \gamma \quad (9)$$

получим систему дифференциальных уравнений линейную по управлениям:

$$\begin{cases} \dot{V} = -g \sin \vartheta + g u_1 \\ \dot{\vartheta} = -\frac{\cos \vartheta}{V} g + \frac{g}{V} u_2 \\ \dot{\psi} = -\frac{g}{V \cos \vartheta} u_3 \\ \dot{H} = V \sin \vartheta \\ \dot{L} = V \cos \vartheta \cos \psi \\ \dot{Z} = -V \cos \vartheta \sin \psi \end{cases} \quad (10)$$

Далее, для удобства применения, представим как систему 3 дифферен-

циальных уравнений 2-ого порядка, положив:

$$y_1 = H, y_2 = L, y_3 = Z, y_4 = \dot{y}_1, y_5 = \dot{y}_2, y_6 = \dot{y}_3$$

получим:

$$\begin{cases} \ddot{y}_1 = -g + u_1 g \sin \vartheta + u_2 g \cos \vartheta \\ \ddot{y}_2 = u_1 g \cos \vartheta \cos \psi - u_2 g \sin \vartheta \cos \psi + u_3 g \sin \psi \\ \ddot{y}_3 = -u_1 g \cos \vartheta \sin \psi + u_2 g \sin \vartheta \sin \psi + u_3 g \cos \psi \end{cases} \quad (11)$$

Замена переменных обратима, так как по следующим формулам можно найти остальные значения.

$$\begin{cases} V = \sqrt{y_4^2 + y_5^2 + y_6^2} \\ \sin \theta = \frac{y_4}{V} \\ \cos \theta = \frac{\sqrt{y_5^2 + y_6^2}}{V} \\ \sin \psi = -\frac{y_6}{\sqrt{y_5^2 + y_6^2}} \\ \cos \psi = \frac{y_5}{\sqrt{y_5^2 + y_6^2}} \end{cases} \quad (12)$$

Стоит отметить, что замена переменных обратима только при выполнении следующих условий:

$$V > 0, |\theta| < \frac{\pi}{2}, -\pi < \psi < \pi$$

Путь полёт БПЛА представляется как его переход из одного состояния в другое с помощью чего и строятся программные управления. Состояние БПЛА представим в следующем виде:

$$\begin{cases} V|_{t=t_0} = V_0, \vartheta|_{t=t_0} = \vartheta_0, \psi|_{t=t_0} = \psi_0, \\ H|_{t=t_0} = H_0, L|_{t=t_0} = L_0, Z|_{t=t_0} = Z_0 \end{cases} \quad (13)$$

На паре таких состояний можно поставить и решить задачу терминаль-

ного управления для формулы (11). Решение будем искать в виде полиномов 5-ой степени на отрезке $[0, 1]$ в виде:

$$y_i(t) = p_i\left(\frac{t - t_0}{\delta}\right), i = \overline{1, 3},$$

где δ - расчетное время маневра. Стоит понимать, что нас интересуют значения полиномов только в отрезке $[0, 1]$, что значит, что будет необходимо проводить следующее преобразование времени:

$$\tau = \frac{t - t_0}{\delta} \quad (14)$$

Начальные значения функций $y_i(t)$ и их производных первого и второго порядка можно найти по формуле (11) и следующей формуле:

$$y = \begin{pmatrix} H \\ L \\ Z \end{pmatrix}, \dot{y} = \begin{pmatrix} V \sin \theta \\ V \cos \theta \cos \psi \\ -V \cos \theta \sin \psi \end{pmatrix} \quad (15)$$

Для поиска коэффициентов данных полиномов воспользуемся следующими формулами:

$$\begin{cases} \alpha_0 = y_{i0} \\ \alpha_1 = \dot{y}_{i0}\delta \\ \alpha_2 = \frac{1}{2}\ddot{y}_{i0}\delta^2 \\ \alpha_3 = 10(y_{i*} - y_{i0}) - 2\delta(3\dot{y}_{i0} + 2\dot{y}_{i*}) - \frac{\delta^2}{2}(3\ddot{y}_{i0} - \ddot{y}_{i*}) \\ \alpha_4 = 15(y_{i0} - y_{i*}) + \delta(8\dot{y}_{i0} + 7\dot{y}_{i*}) + \frac{3\delta^2}{2}(3\ddot{y}_{i0} - 2\ddot{y}_{i*}) \\ \alpha_5 = 6(y_{i*} - y_{i0}) - 3\delta(\dot{y}_{i0} + \dot{y}_{i*}) - \frac{\delta^2}{2}(\ddot{y}_{i0} - \ddot{y}_{i*}) \end{cases} \quad (16)$$

Необходимо отметить, что изменение времени маневра δ может сильно повлиять на полученную траекторию. В [4] предлагается несколько способов выбора времени маневра, однако также отмечено, что выбор данной величины носит эмпирический характер. Так как разные методы выбора времени предполагают, что могут оказаться некоторые ограничения на переменные состояния и управления. В данной работе выбор величины вре-

мени выполняется эмпирически.

Для нахождения необходимых программных управлений необходимо решить систему (11) относительно управлений. В результате получим:

$$\begin{cases} u_1 = \sin \vartheta + \frac{\ddot{y}_1 \sin \vartheta + \ddot{y}_2 \cos \vartheta \cos \psi - \ddot{y}_3 \cos \vartheta \sin \psi}{g} \\ u_2 = \cos \vartheta + \frac{\ddot{y}_1 \cos \vartheta - \ddot{y}_2 \sin \vartheta \cos \psi + \ddot{y}_3 \sin \vartheta \sin \psi}{g} \\ u_3 = \frac{\ddot{y}_2 \sin \psi + \ddot{y}_3 \cos \psi}{g} \end{cases} \quad (17)$$

После получения программных управлений можно вычислить управления в форме обратных связей, для этого воспользуемся следующей формулой:

$$u(y, \dot{y}) = \frac{1}{g^2} B^T(y, \dot{y}) B(y_p, \dot{y}_p) u_p - \frac{1}{g^2} B^T(y, \dot{y}) (K_1(\dot{y} - \dot{y}_p) + K_2(y - y_p)), \quad (18)$$

где u_p, y_p, y, u - соответственно вектора программных и текущих траекторий и управлений, K_1, K_2 - диагональные матрицы с элементами соответственно k_1, k_2 , которые в дальнейшем будем называть коэффициентами стабилизации, а матрица B задана следующим образом:

$$B(y, \dot{y}) = g \begin{pmatrix} \sin \vartheta & \cos \vartheta & 0 \\ \cos \vartheta \cos \psi & -\sin \vartheta \cos \psi & \sin \psi \\ -\cos \vartheta \sin \psi & \sin \vartheta \sin \psi & \cos \psi \end{pmatrix}$$

В итоге получаем необходимые управления для движения БПЛА по заданной траектории.

6 Описание алгоритма

Приведем алгоритм удержания БПЛА на расчетной траектории:

1. На первом шаге необходимо инициировать начальные и конечные состояния объекта, которые соответствуют необходимому маневру;

2. На втором шаге необходимо перейти к линейной по управлениям системе, для чего терминальные состояния объекта рассчитываются по (15) и (11);
3. С помощью формулы (16) вычислить коэффициенты трех полиномов и получаем расчетную траекторию движения БПЛА;
4. Далее вычислить время по формуле (14) и найти векторы $y_p(t)$, $\dot{y}_p(t)$ - соответствующие программным управлениям;
5. С помощью полученных векторов состояний по формуле (17) найти вектор $u_p(t)$ программных управлений;
6. Зная текущее состояния летательного аппарата, определяемое величинами H, L, Z, V, θ, ψ с учетом (10) получить векторы y, \dot{y} по формуле (15);
7. Используя программное и текущее состояние, и, кроме того, вектор программных управлений с помощью формул (18) вычислить управление в форме обратной связи;
8. С помощью полученного управления по формулам (11) получить новое состояние БПЛА соответствующее текущему моменту времени;
9. Перейти к шагу 4) и вычислить новое время. Кроме случая когда выделенное для маневра время закончилось.

Далее приведем описание алгоритма в форме блок-схемы. Такой вид описания алгоритма оказывается наиболее удобным при дальнейшей реализации его с помощью пакета MATLAB/Simulink и изображен на рис. 2. В особенности из-за того, что программы с использованием Simulink представляются в похожем виде.

Кратко опишем функциональность каждого блока.

- В блоке «Исходные данные» содержатся исходные данные для маневра, позволяющие поставить терминальную задачу. Это два терминальных состояния в виде формул (13), (9) и времени t ;
- В блоке «Расчет коэффициентов полиномов» по данным из блока "Исходные данные" происходит расчет коэффициентов полиномов по



Рис. 2: Общая схема алгоритма

формуле (16);

- В блоках «Расчет коэффициентов производной полинома первого порядка» и «Расчет коэффициентов производной полинома второго порядка» происходит вычисление коэффициентов полиномов, которые являются соответственно первой и второй производной исходного полинома;
- В блоке «Генератор времени» происходит генерация времени согласно формуле (14). Выходным параметром этого блока является сгенерированные по данной формуле значения;
- В блоках «Вычисление значения полинома» соответственно названию происходит вычисление значений полиномов. Входные параметры - коэффициенты и время, выходные - значение полинома в данной точке;
- В блоке «Расчет скорости и углов» по значения первых производных полиномов по формулам (12) находим значения скорости и углов, которые описывают состояние летательного аппарата в данный момент времени;
- В блоке «Расчет программных управлений» согласно отношениям (17) вычисляются программные управления. Входные данные - угол наклона траектории, угол пути и значения второй производной полинома в текущий момент времени, выходные вектор программных

управлений;

- В блоке «Расчет управлений» по формуле (18) вычисляются управления в форме обратных связей. Входные данные - значения полинома и значение его первой производной, угол наклона траектории, угол пути, вектор программных управлений и вектора Y, \dot{Y} , описывающие текущее состояние летательного аппарата. Выходные данные управления в форме обратных связей;
- В блоке «Расчет нового положения летательного аппарата» происходит двойное интегрирование формулы (11) после чего мы получаем новое положение в пространстве нашего летательного аппарата, описываемое вектором Y, \dot{Y} . Входные данные - вектор управлений и значения углов наклона траектории и пути, выходные данные - вектора состояния БПЛА Y, \dot{Y} ;
- В блоке «Построение пространственных траекторий» происходит сбор данных о перемещении БПЛА в пространстве и дальнейшее построение соответствующих графиков. Входные данные - положение летательного аппарата в пространстве.

7 Реализация алгоритма

7.1 Описание использованных средств

Для реализации алгоритма был использован пакет математического моделирования Matlab совместно с графической средой имитационного моделирования Simulink. Для графического отображения результатов моделирования использовался пакет 3D Animation, поставляемый вместе с Simulink.

MATLAB - это мультипарадигмальная среда для проведения численных вычислений, которая изначально была ориентирована на работу с матрицами. В настоящее время кроме базовых функций, таких как различные преобразования матриц, построение графиков и взаимодействия с другими языками (C, C++, Java, Fortran, Python), предоставляет большое специализированных пакетов (Toolbox) для таких решение различных задач из

областей финансового анализа, баз данных, нечеткой логики, нейронных сетей и других.

Важным элементом пакета MATLAB без сомнения является специализированный пакет программных средств Simulink, так как данный инструмент позволяет достаточно просто моделировать динамические объекты и процессы. Одним из плюсов данного инструмента является наглядность механизма функционирования модели - система представляется в виде некой блок-схемы, которая составлена из различных функциональных блоков.

3D Animation toolbox - это библиотека для создания 3D миров и анимаций внутри них. Объекты в ней представляются с помощью языка-стандарта 3D моделирования - Virtual Reality Modeling Language (VRML). Важным фактом является то, что средства Simulink представляют инструменты для непосредственной связи анимации и имитационной модели. Поэтому для визуализации какого-нибудь процесса достаточно создать виртуальный мир с интересующими нас объектами в специальном редакторе миров 3D World Editor и после этого добавить специальные блоки к исходной модели реализованной на Simulink для изменения необходимых параметров объектов, таких как расположение, ориентация, масштабирование и другие.

7.2 Результаты моделирования

В результате практической реализации алгоритма была получена программа, которая моделирует удержание БПЛА на расчетной траектории. На рис. 3 приведен графический облик этой программы написанный в среде Simulink. Несложно заметить, что данная программа в общем представляет собой практическую реализацию согласно описанию на рис. 2.

Для проверки получившейся программы проведем несколько вычислительных экспериментов. Для начала промоделируем такой маневр БПЛА как разворот. Необходимые для данного маневра исходные данные представлены в таблице 1.

Приведем дальше результаты моделирования полёта летательного ап-

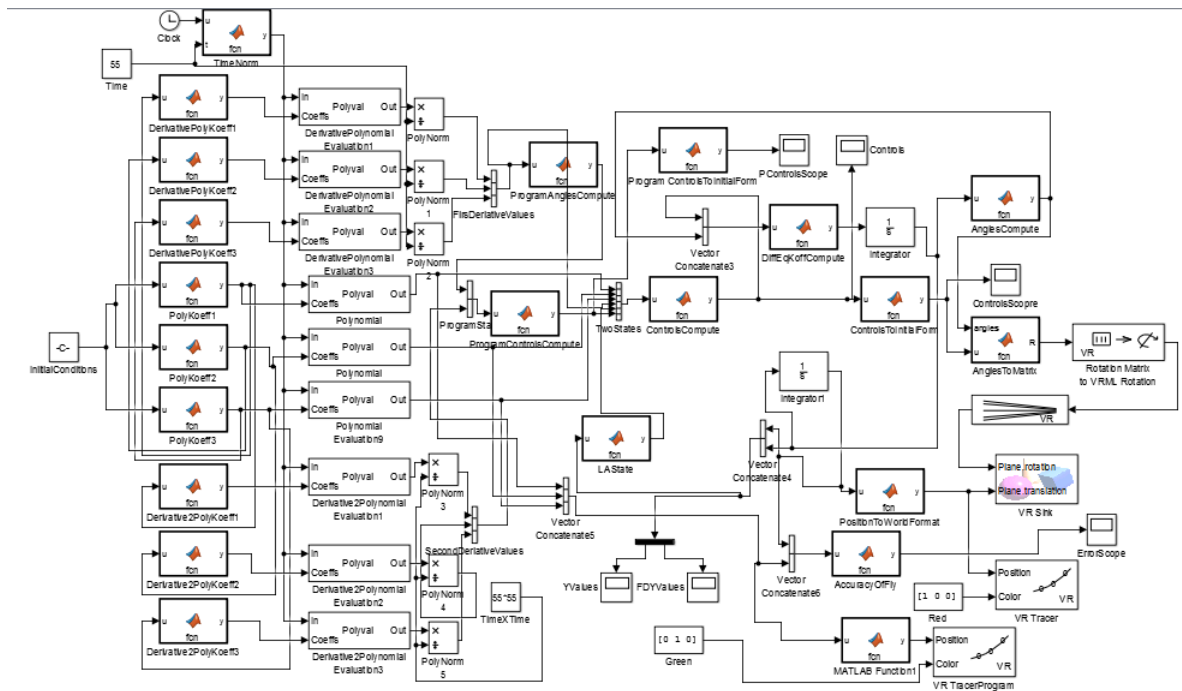


Рис. 3: Представление программы.

Таблица 1: Данные для выполнения маневра разворота.

	V, м/с	θ , град	ψ , град	H, м	L, м	Z, м	n_x	n_y	γ , град
$t_0 = 0$ сек	50	0	0	1000	0	0	0	1	0
$t_* = 55$ сек	50	0	180	1200	700	-550	0	1	0

парата. Графики изменения положения в пространстве от времени полёта показаны на рис. 4. На рис. 5 показан график изменения углов, а на рис. 6 график изменения скорости полёта. Из графиков видно, что были соблюдены начальные и конечные условия задачи. График расхождения полученной траектории и расчетной приведен на рис. 7. По этим данным можно понять, что ошибка управления по каждой координате достаточно мала и сходиться к нулю при $t \rightarrow +\infty$. Графики управлений движением БПЛА приведены на рис. 8. На рис. 9 приведена траектория по которой пролетел БПЛА во время моделирования с использованием анимации полёта. Красная траектория соответствует реальной траектории полёта, синяя - программной.

Необходимо отметить, что данные результаты были получены при коэффициентах стабилизации $k_1 = k_2 = 0.01$. Рассмотрим результаты моделирования при других коэффициентах. График на рис. 10 получен при использовании коэффициентов стабилизации 0.001, 0.001. Можно заметить,

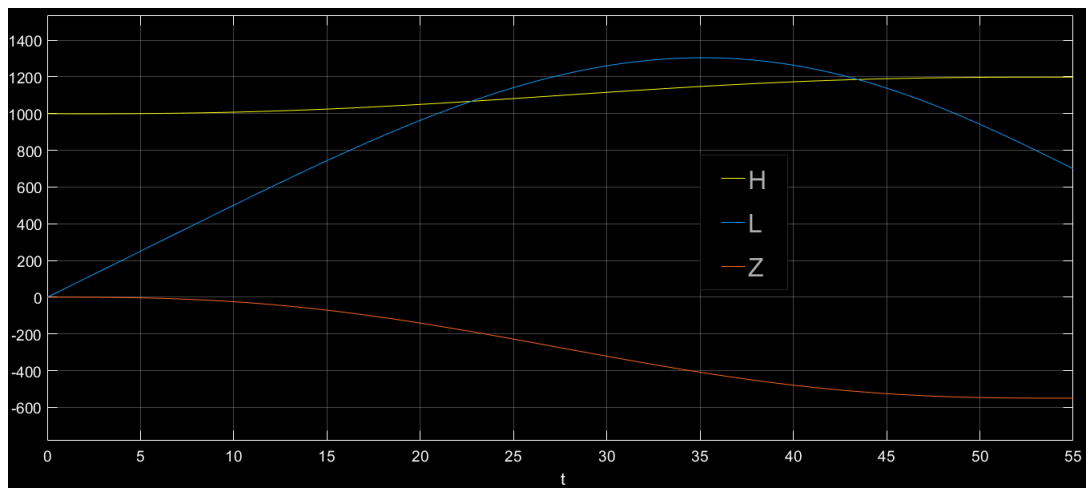


Рис. 4: График изменения координат во времени.

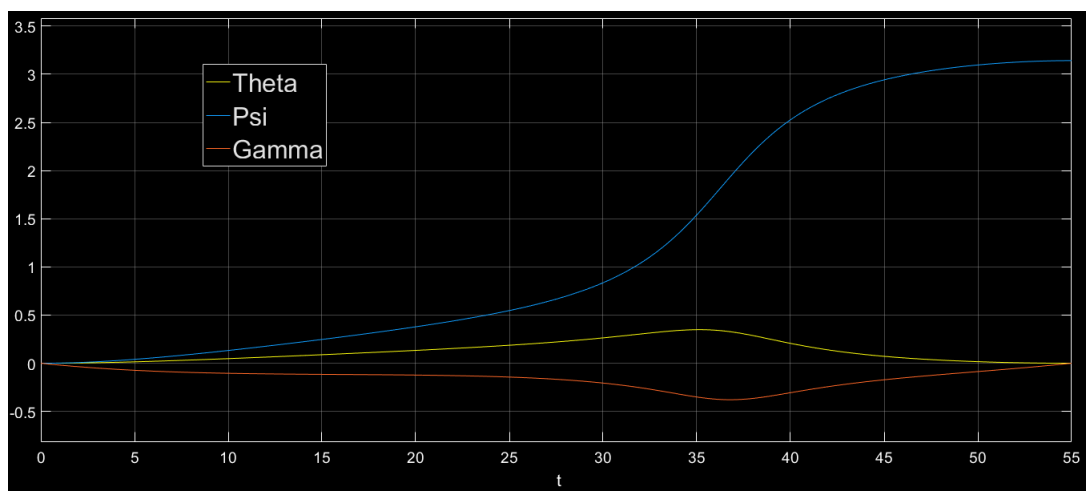


Рис. 5: График изменения углов во времени.

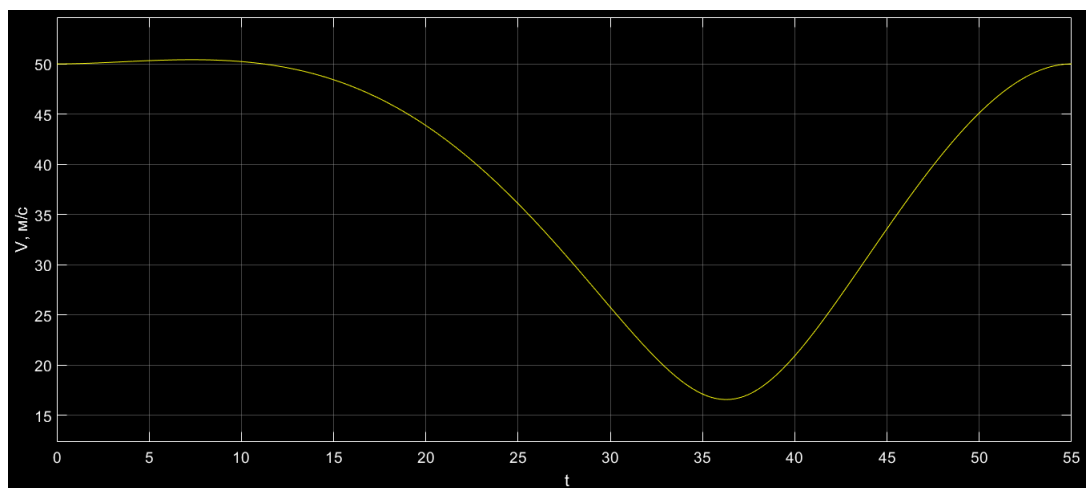


Рис. 6: График изменения скорости во времени.

что ошибка управления увеличилась. На рис. 11 изображен график разницы программной и реализованной траектории при коэффициентах $-0.1, -0.1$.

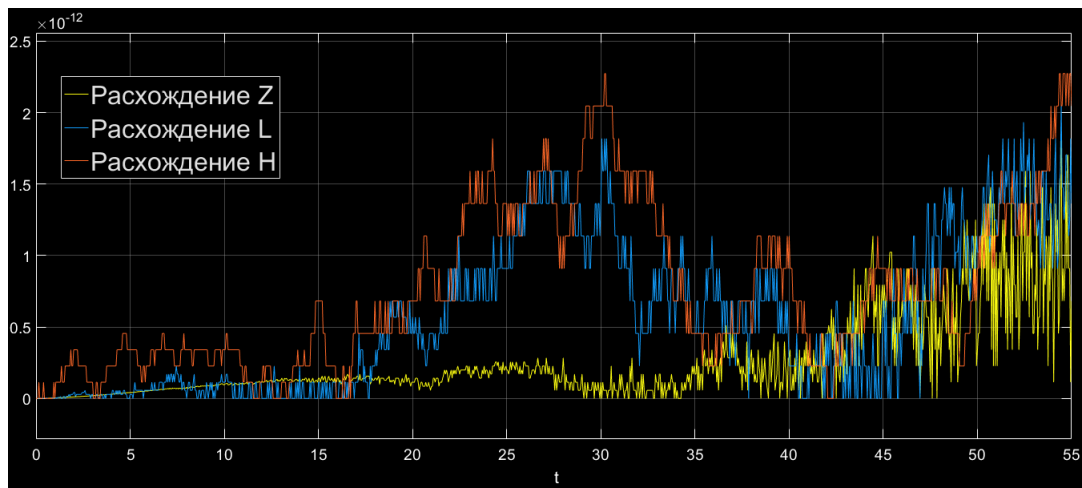


Рис. 7: График изменения ошибки траектории полёта во времени.

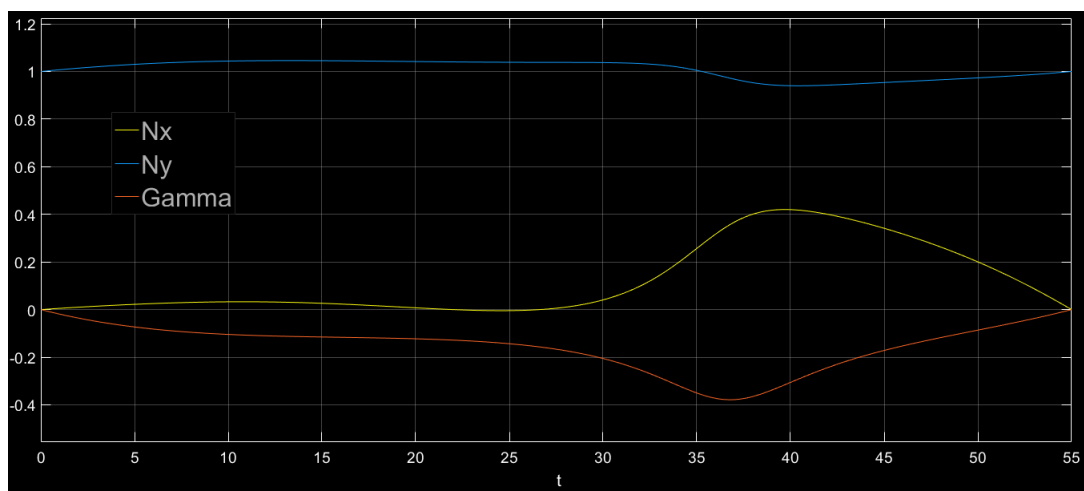


Рис. 8: График изменения управлений во времени.

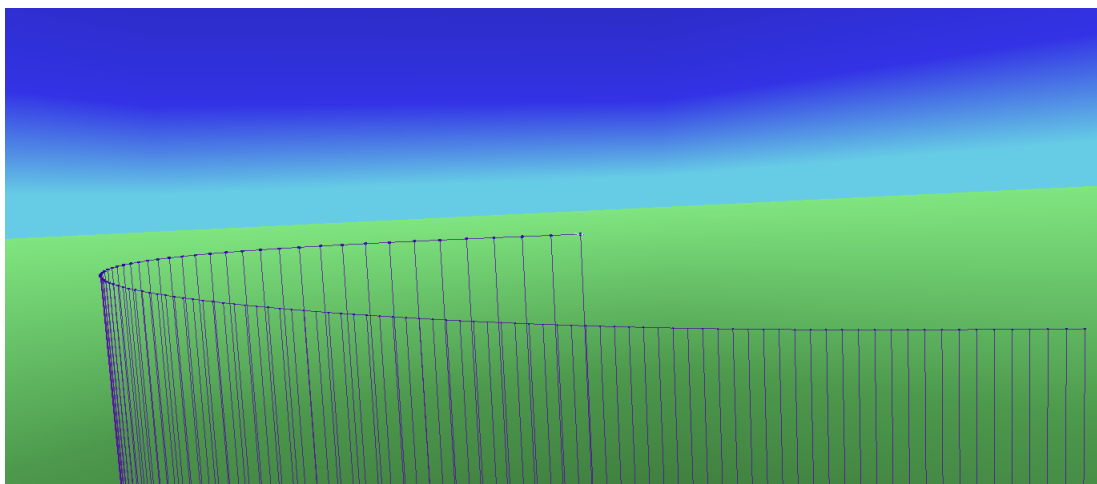


Рис. 9: Траектория полёта согласно данным анимации

Как видно из графика полученное решение расходиться. Что подтверждает теоретические рассуждения о необходимой положительности коэффициен-

тов стабилизации.

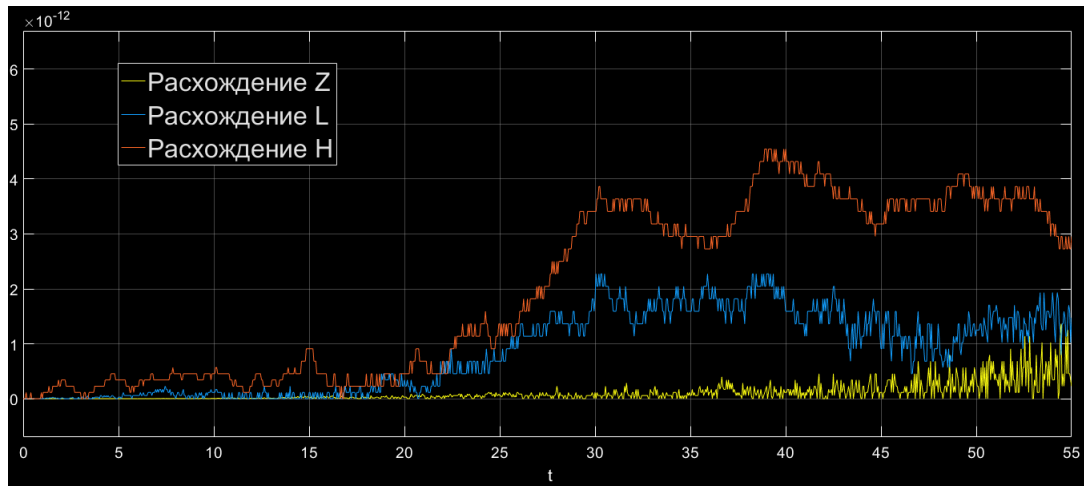


Рис. 10: График изменения ошибки траектории полёта во времени.

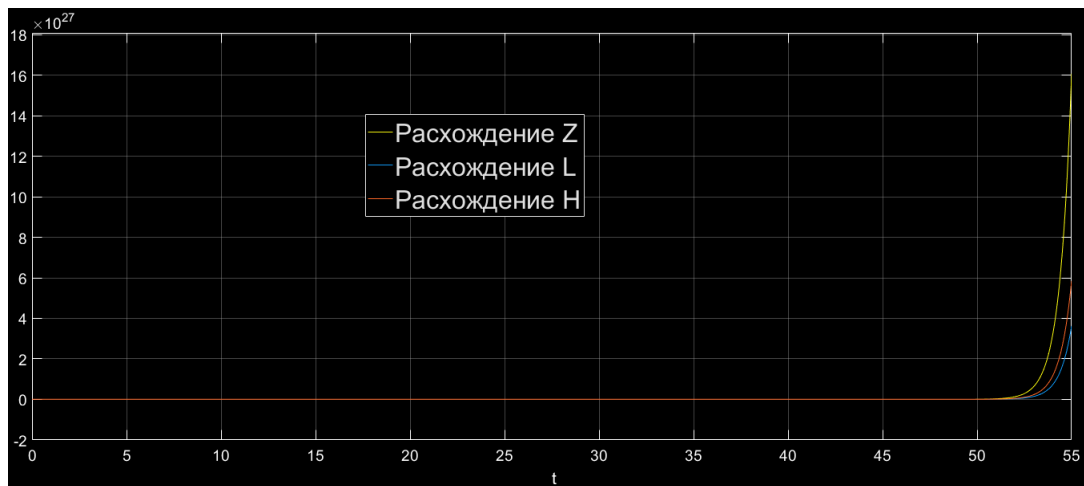


Рис. 11: График изменения ошибки траектории полёта во времени.

Второй промоделированный маневр БПЛА будет набор высоты. Необходимые для данного маневра исходные данные представлены в таблице 2.

Таблица 2: Данные для выполнения маневра набора высоты.

	V, м/с	θ , град	ψ , град	H, м	L, м	Z, м	n_x	n_y	γ , град
$t_0 = 0$ сек	30	15	0	1000	0	0	0	1	0
$t_* = 40$ сек	50	0	0	1500	1500	100	0	1	0

После моделирования полёта БПЛА по таким данным получены следующие результаты. На рис. 12 изображен график изменения координат

объекта от времени. На рис. 13 приведен график ошибки управления по каждой координате в каждый момент времени. На рис. 14, 15 показаны графики изменения углов и скорости во времени соответственно. Графики управлений движением БПЛА приведены на рис. 16. На рис. 17 изображена траектория - результат анимации полёта модели.

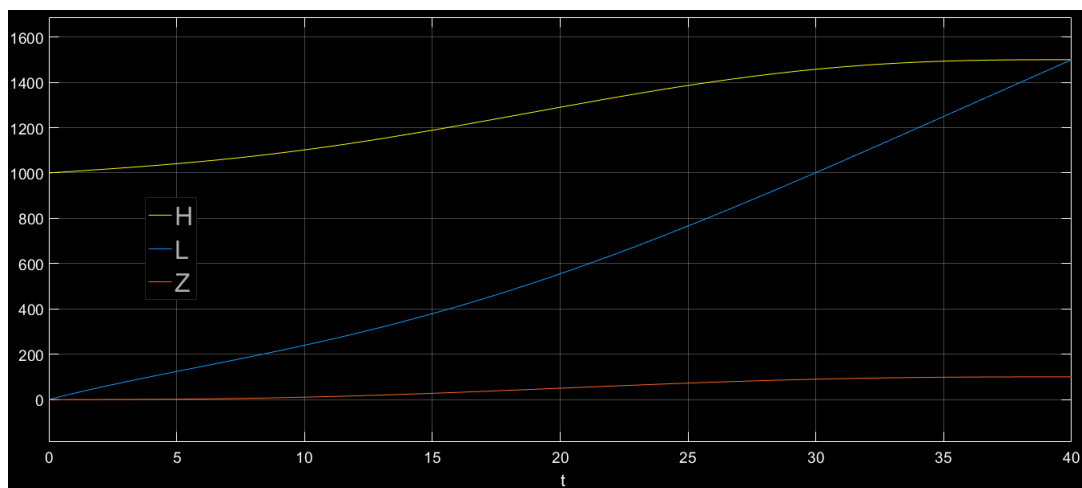


Рис. 12: График изменения координат во времени.

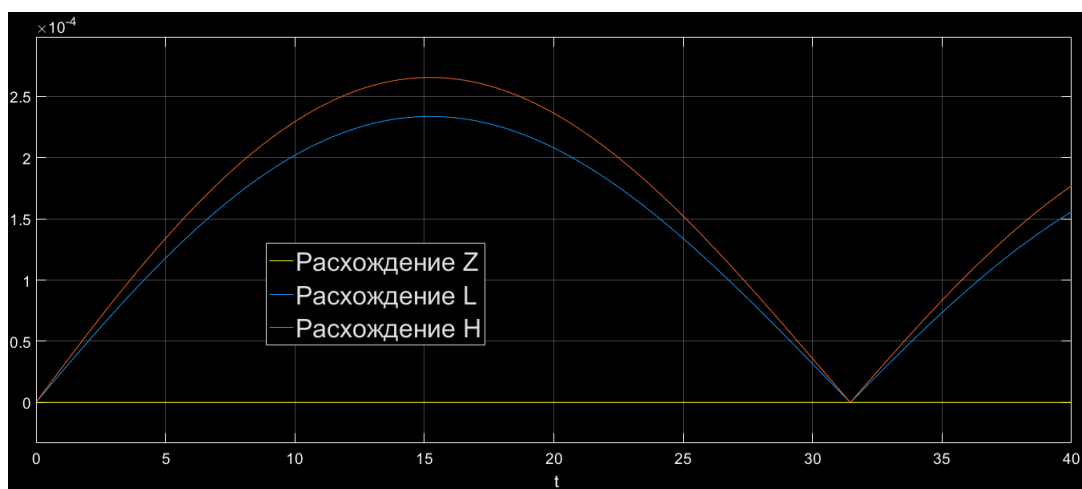


Рис. 13: График изменения ошибки траектории полёта во времени.

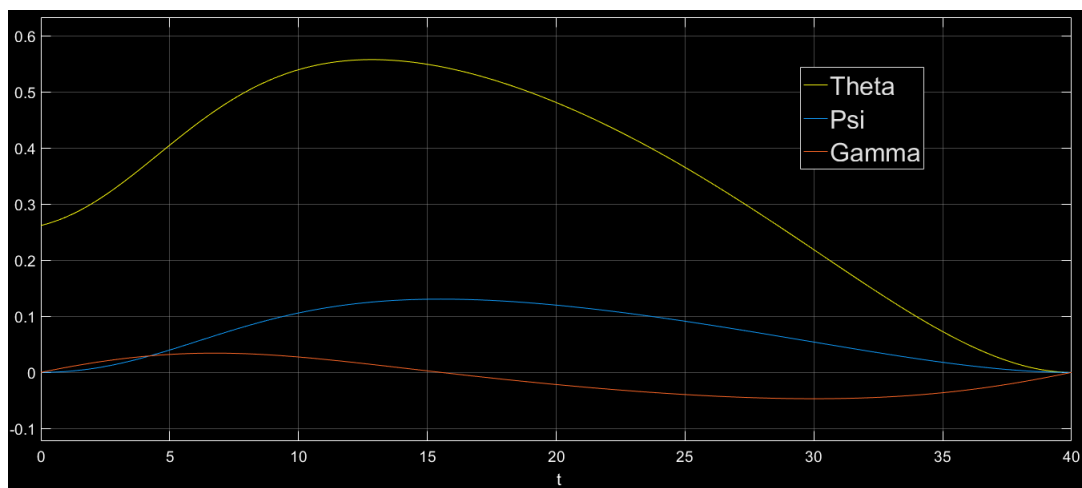


Рис. 14: График изменения углов во времени.

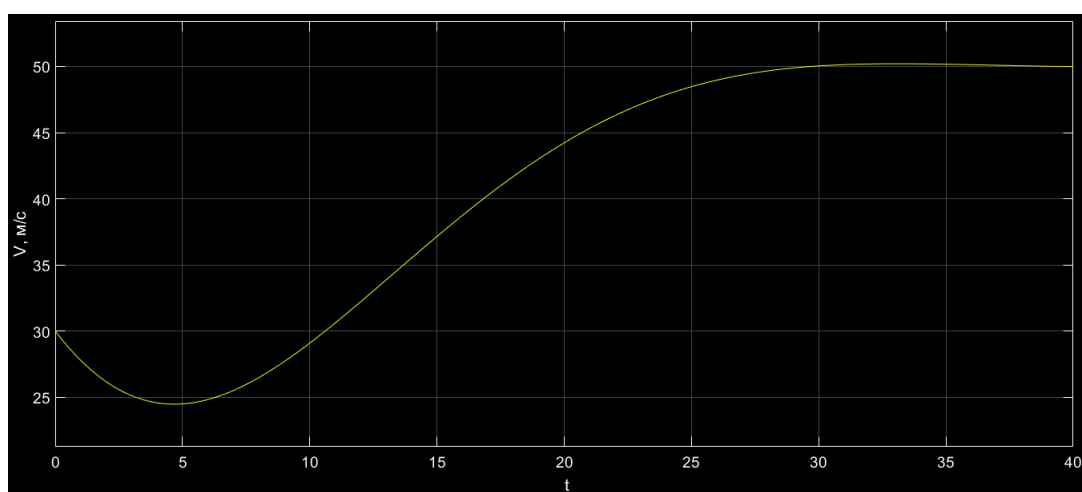


Рис. 15: График изменения скорости во времени.

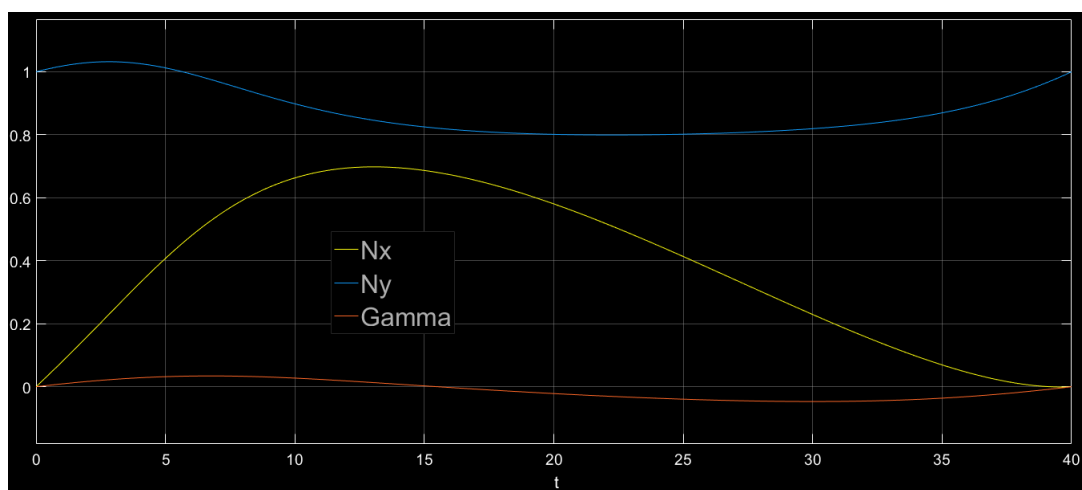


Рис. 16: График изменения управлений во времени.

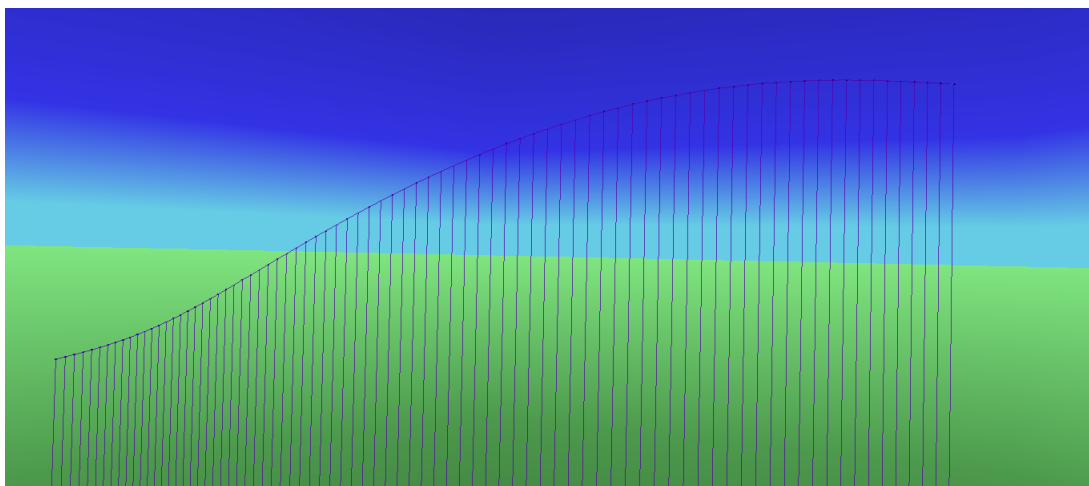


Рис. 17: Траектория полёта согласно данным анимации

8 Выводы

1. Показана актуальность создания автономных БПЛА, как более совершенных устройств;
2. Приведено теоретическое описание задач поиска управлений. Указаны причины использования поиска управлений на основе концепции обратных задач динамики;
3. Рассмотрена математическая модель движения БПЛА, как система нелинейных дифференциальных уравнений первого порядка;
4. Описан алгоритм построения расчетной траектории на основе полиномов пятой степени, удовлетворяющих граничным условиям;
5. Описан метод нахождения управлений, которые обеспечивают асимптотическую устойчивость движения БПЛА по заданной траектории;
6. В рамках созданного программного комплекса синтезирован и реализован алгоритм удержания БПЛА на расчетной траектории. Программный комплекс позволяет моделировать динамическую модель движения БПЛА и получить графики фазовых координат и необходимых управлений.

9 Заключение

В данной работе основное внимание было уделено созданию и реализации алгоритма удержания БПЛА на траектории близкой к расчетной. Дано краткое описание существующего устройства управления БПЛА и его недостатков, определена необходимость создания автоматически управляемых БПЛА. Реализацию управлений предполагается осуществлять исходя из принципов обратной задачи динамики и управления с обратными связями. Описана математическая модель, основанная на данных принципах, и реализована предложенная компьютерная модель с использованием технологий MATLAB/Simulink.

Дальнейшее изучение данной темы предполагается вести в следующих направлениях:

- Изучение возможности реализации более сложных траекторий. В настоящей версии программы накладываются достаточно серьезные ограничения на траекторию, так как любая траектория задается полиномом 5-ой степени;
- Учет времени при выборе маневра БПЛА;
- Создание версии программы с ограничениями, накладываемыми на движение БПЛА в соответствии с реально существующими беспилотниками.

10 Список литературы

1. Канатников А.Н., Шмагина Е.А. Задача терминального управления движением летательного аппарата // Нелинейная динамика и управление, 2010. Вып.7 С. 79-94
2. Крутько П.Д. Обратные задачи динамики управляемых систем. Нелинейные модели. М.: Наука, 1988. С. 9-18
3. Крутько П.Д. Обратные задачи динамики в теории автоматического управления. М.: Машиностроение, 2004. С. 406
4. Канатников А.Н., Крищенко А.П., С.Б. Ткачев Допустимые пространственные траектории беспилотного летательного аппарата в вертикальной плоскости // Наука и образование, 2012
5. Андрющенко Т.А. Разработка системы управления беспилотным летательным аппаратом в режиме «зависание» // НГУ, 2013
6. Официальный сайт разработчика MATLAB <http://www.mathworks.com>
7. Маркеев А.П. Теоретическая механика. М.: Наука, 1990. С. 380

А Приложение

Приведем данные анимации полёта. Для маневра разворота на рис. 18 - 24. Для маневра набора высоты на рис. 25 - 30.

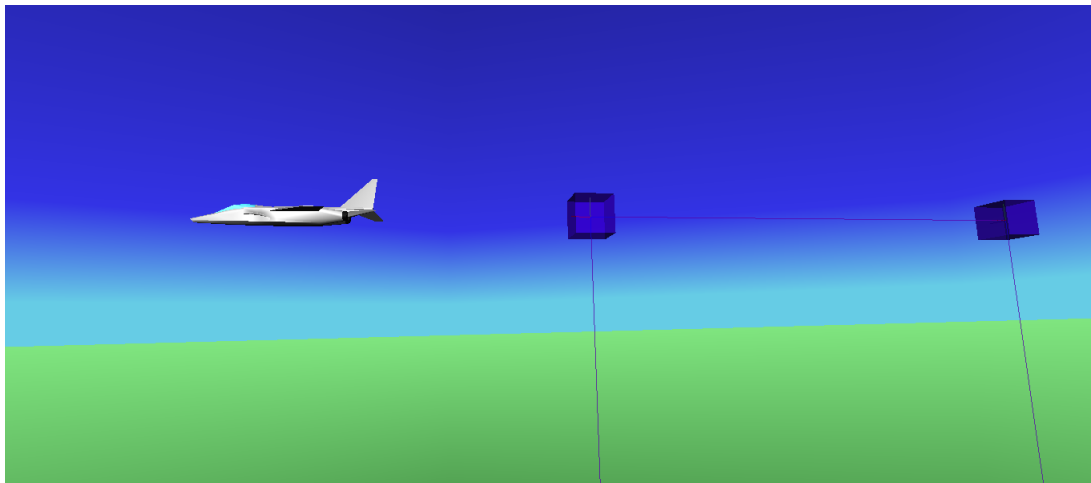


Рис. 18: Анимация полёта. Маневр разворота. Часть 1.

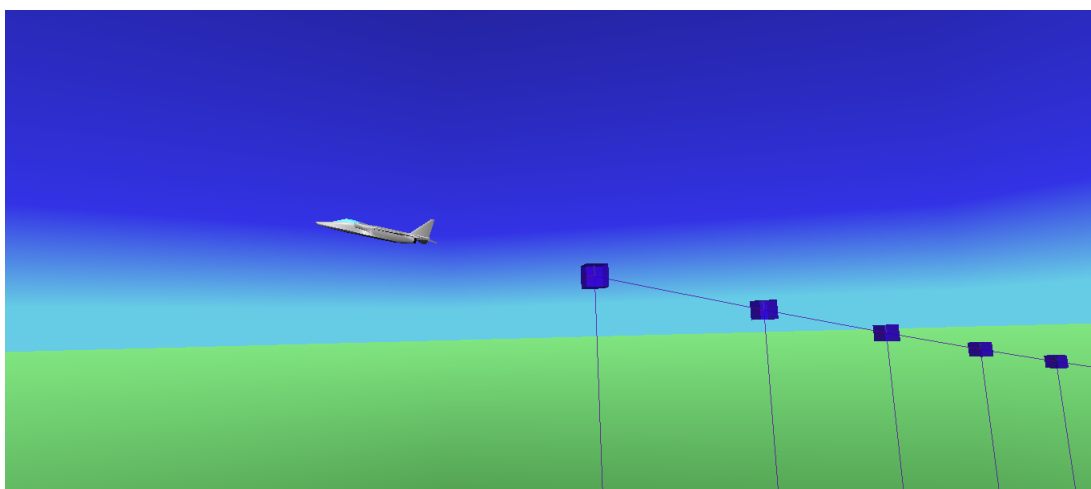


Рис. 19: Анимация полёта. Маневр разворота. Часть 2.

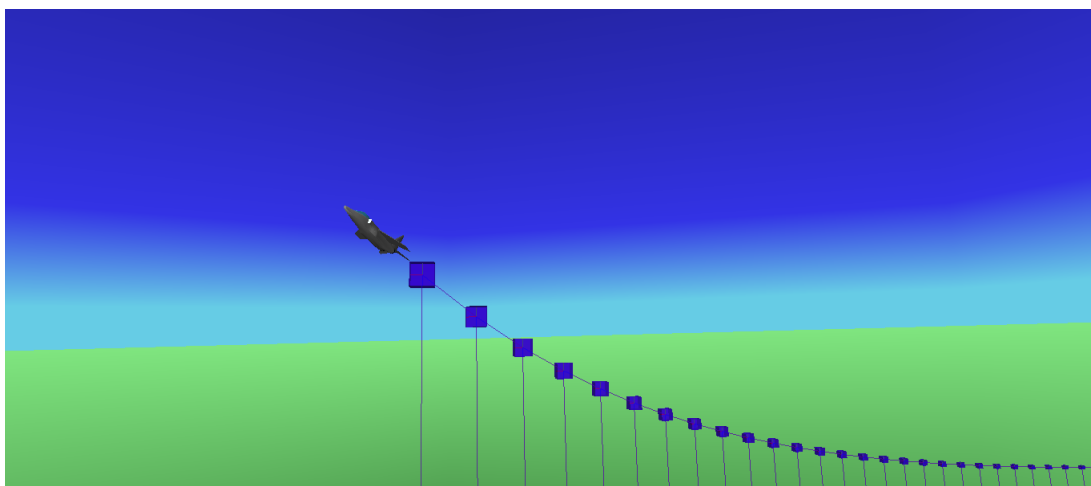


Рис. 20: Анимация полёта. Маневр разворота. Часть 3.

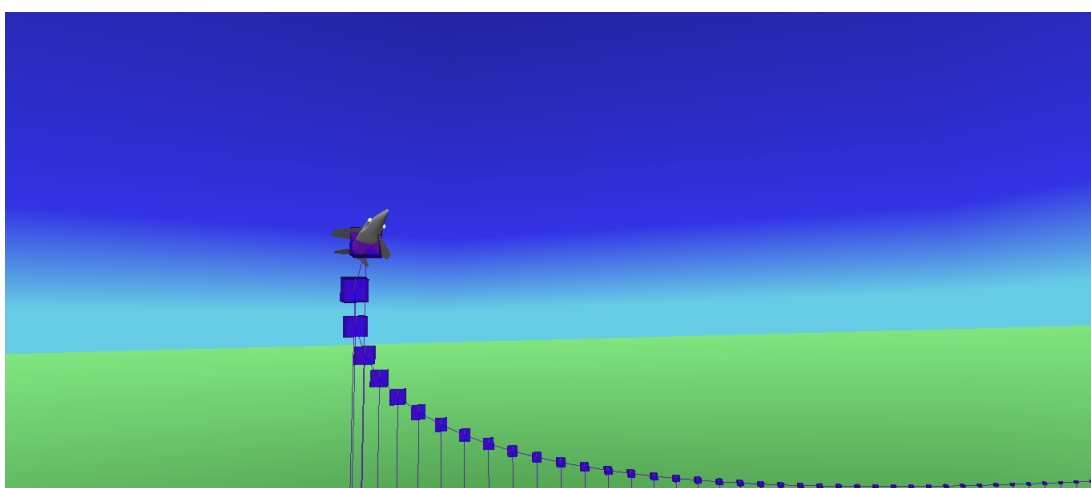


Рис. 21: Анимация полёта. Маневр разворота. Часть 4.

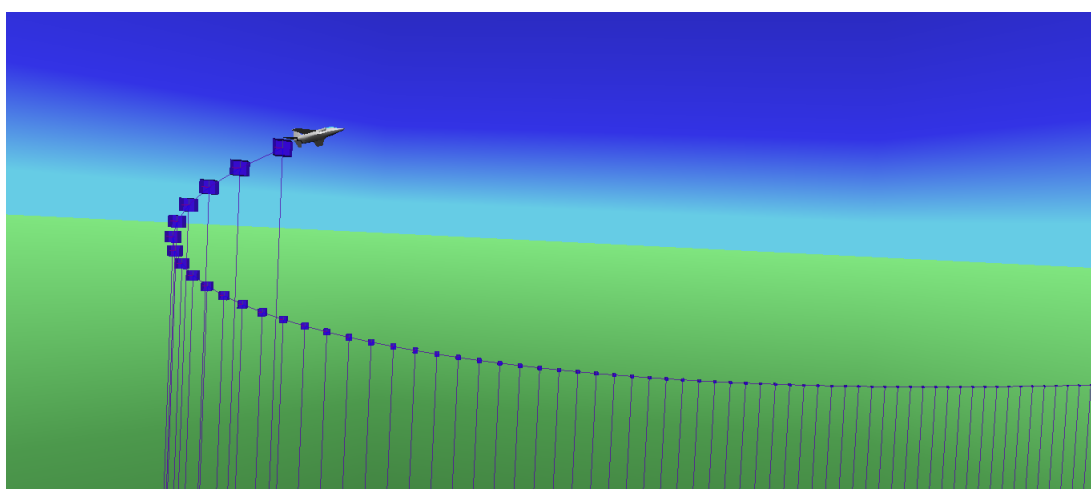


Рис. 22: Анимация полёта. Маневр разворота. Часть 5.

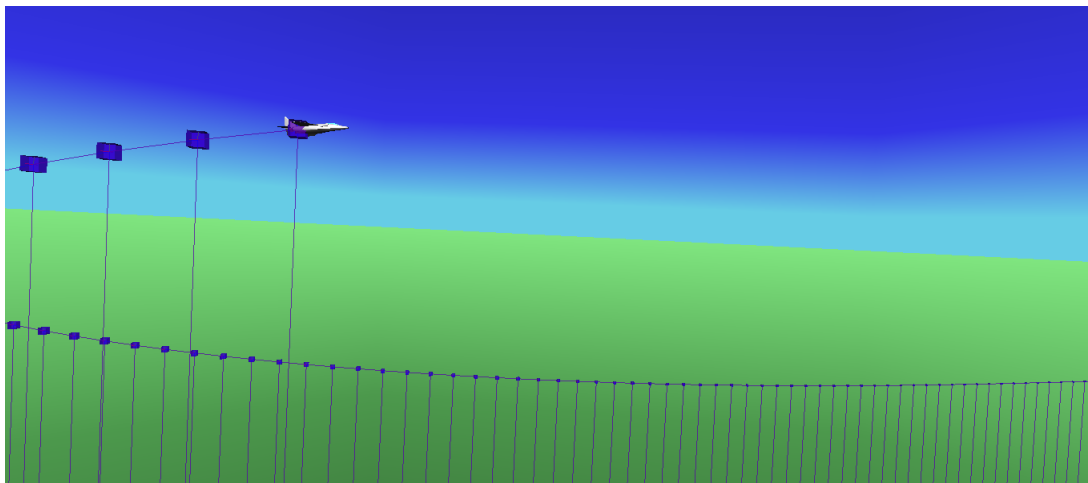


Рис. 23: Анимация полёта. Маневр разворота. Часть 6.

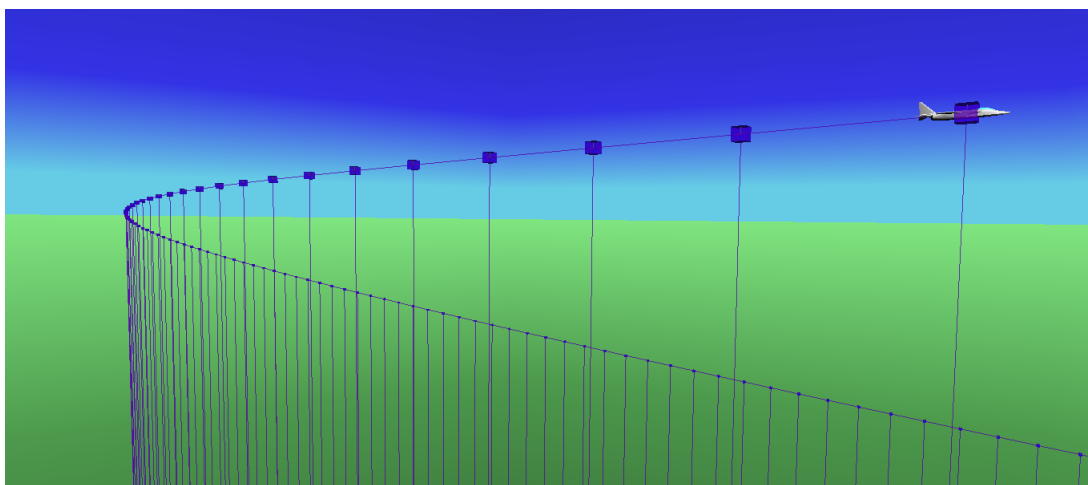


Рис. 24: Анимация полёта. Маневр разворота. Часть 7.

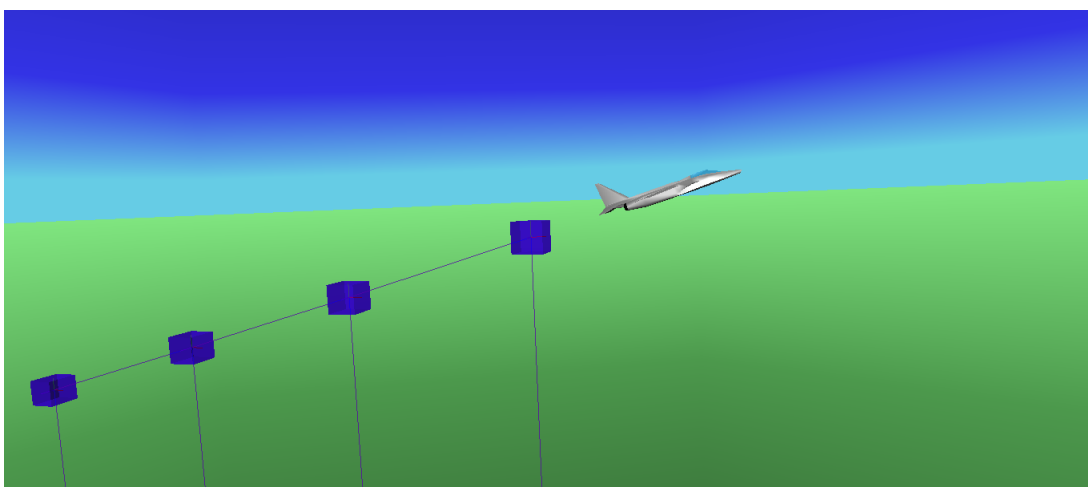


Рис. 25: Анимация полёта. Маневр набора высоты. Часть 1.

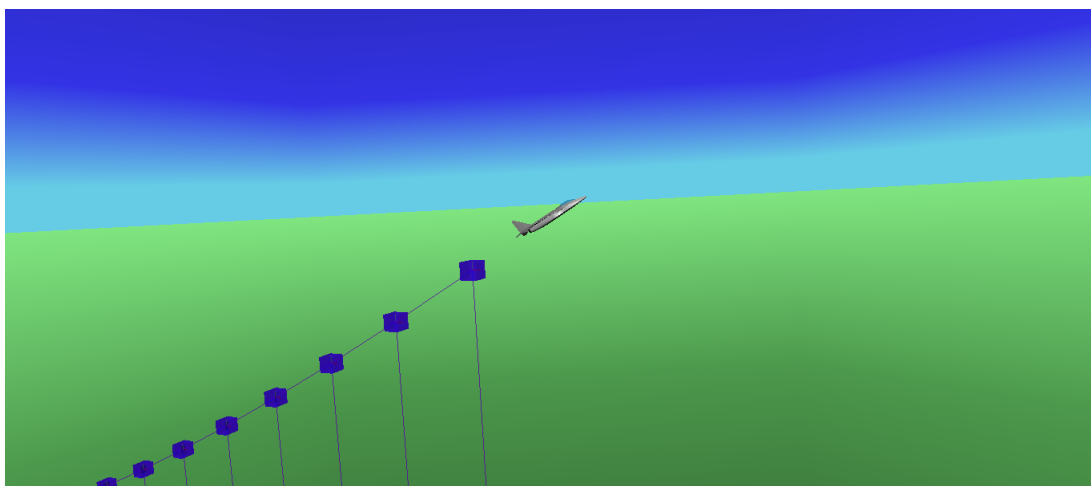


Рис. 26: Анимация полёта. Маневр набора высоты. Часть 2.

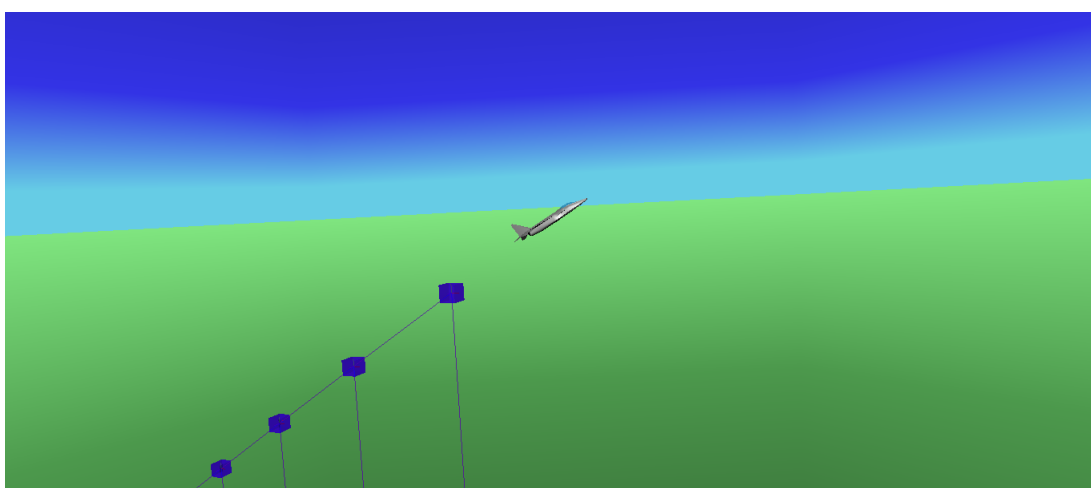


Рис. 27: Анимация полёта. Маневр набора высоты. Часть 3.

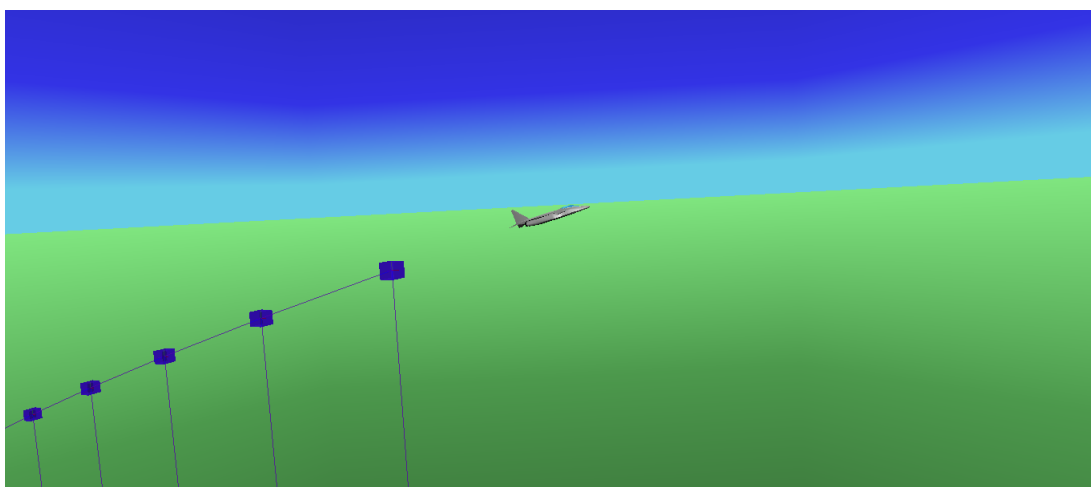


Рис. 28: Анимация полёта. Маневр набора высоты. Часть 4.

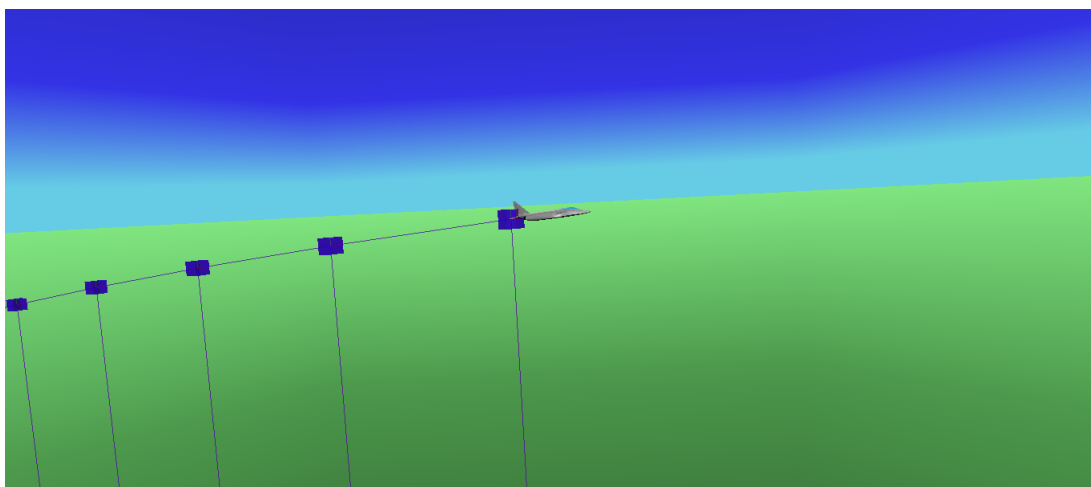


Рис. 29: Анимация полёта. Маневр набора высоты. Часть 5.

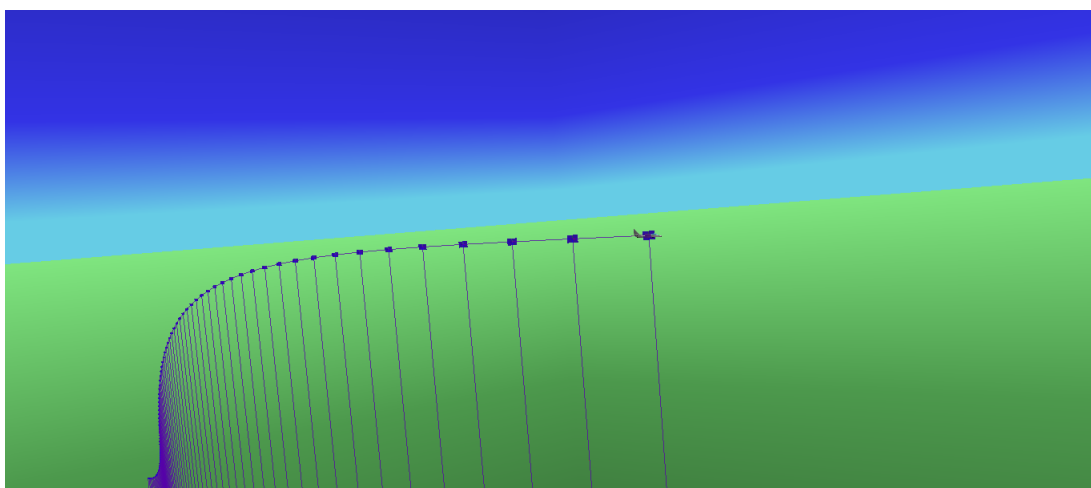


Рис. 30: Анимация полёта. Маневр набора высоты. Часть 6.